

Exercise Sheet 2

Deadline: May 14, 2024 – 04:00 am CEST

Exercise 1 (Project "Room Manager")

The administration of a university is asking us for help. Currently their room management (listing available rooms and its capacities, adding and clearing reservations. . .) is not sufficiently efficient. In fact, the various secretaries of the faculties maintain local replicas of a spreadsheet listing all rooms, their properties, and current reservations. These replicas need to be synchronized manually between the secretaries and are shared via email. This is a cumbersome and error-prone task and should be replaced with a more modern solution.

The university's administration found an open-source library called *roommanager* which provides the required functionality to tackle this problem. However, only the library itself is provided by the maintainers of this project, but no user interface. It is now your job to implement a basic UI for the library.

Your repositories already contain the template for this in the `02` subdirectory. Since the library is hosted in a git repository as well, we can use a *git submodule*. (This is basically a symbolic link inside your repository to another git repository.) In order to retrieve the content from this *submodule* you need to call first `git submodule init` and then `git submodule update`. (*Note: Typically no further calls to `git submodule` should be necessary.*) Now, you should be able to build the provided (initially almost empty) application which is linked against the *roommanager* library by calling `make`.

You will find the source code for this application skeleton in the `src` directory. The directory `libsrc` should contain the *roommanager* library. All your changes should happen in the `src` directory. No changes in `libsrc` are required.

Calling `make` will automatically build the *roommanager* library first. The library provides an API documentation which can be compiled into HTML documents with the tool `doxygen`¹. A precompiled version of the documentation can be found at <https://teaching.dahahm.de/roomman/>. The *Makefile* contains additional targets to build the library only, to build the documentation, or to execute the tests.

¹<http://www.doxygen.org>

Implement the following functionalities called via the given command line arguments:

1. `./roommanager -l`
Prints a list of all available rooms (including the building name, its capacity, and current reservation state) as a table.
2. `./roommanager -l BCN`
Prints a list of all rooms in building "BCN".
3. `./roommanager -n BCN 321`
Creates an entry for a new room "321" in building "BCN" using the default capacity of 10.
4. `./roommanager -n "Gebaeude 1" 234 32`
Creates an entry for a new room "234" in building "Gebaeude 1" using a capacity of 32.
5. `./roommanager -o BCN 321 20`
Updates the number of persons currently occupying the room "321" in building "BCN" to 20.
6. `./roommanager -c BCN 321 40`
Updates the room capacity for room "321" in building "BCN" to 40.
7. `./roommanager -t "Gebaeude 1" 234`
Toggles the reservation state of "234" in building "Gebaeude 1". If the room was previously reserved it should now be listed as free, if the room was previously free it should now be reserved.
8. `./roommanager -d "Gebaeude 1" 234`
Deletes the room "234" in building "Gebaeude 1" from the database entirely.

Hints:

- Erroneous or missing user input should be intercepted and return an error message. Your program should print a short usage description ("RTFM text").
- Wrong parameters (e.g., too long names for a building or a room) are rejected by the library. In these cases your application should print a qualified error message.
- Each combination of building and room name can only exist exactly once. All tries to add the same combination again, will be rejected by the library and shall trigger a qualified error message in your application.
- Rooms can only be reserved if you specify a username and if the room is currently not used.

- Use the function `strtol(3)` to convert an integer from a string representation into an `int` value.
- A set of tests is provided in the `tests` directory. They can be executed via the *Makefile* (using `make test`) or manually via `sh tests/testroomman.sh -n`. By omitting the `-n` parameter the tests will run interactively and show you a *diff* between the expected output and the output of your implementation.